

Amendments to the Claims

1 Claim 1 (currently amended): A computer-implemented method of programmatically building
2 queries, comprising steps of:

3 programmatically identifying, for a content source, at least one element thereof; each
4 thereof by programmatically obtaining one or more tag names from a markup language document
5 used for rendering a view of the content source;

6 using at least one of the programmatically-obtained tag names programmatically-identified
7 element comprising to consult a lookup component to obtain at least one [[a]] candidate query
8 parameter name for querying the content source; and

9 displaying [[the]] each obtained candidate query parameter(s) parameter name on a user
10 interface display configured ~~to allow a user~~ to build a query command ~~to query the content~~
11 source, responsive to selection by a user of wherein the user can select at least one of the
12 displayed candidate query parameter(s) parameter name or names, to query the content source, to
13 ~~build the query command.~~

Claim 2 (cancelled)

1 Claim 3 (currently amended): The method according to Claim 1, wherein the using
2 programmatically identifying step further comprises ~~the step of consulting a lookup table,~~ using
3 information regarding the user when consulting the lookup component, ~~to thereby determine~~ at
4 least one element usable as a candidate query parameter for the user to select when building the
5 ~~query command to query the content source.~~

1 Claim 4 (currently amended): The method according to Claim 1, further comprising the step of:
2 programmatically identifying at least one query extension parameter name for the query
3 command, responsive to a request from the user to extend ~~add at least one query parameter to the~~
4 query command; and
5 wherein the displaying step further comprises also displaying each of the at least one
6 programmatically-identified query extension ~~parameters~~ parameter name or names as additional
7 ones of the candidate query ~~parameters~~ parameter names.

1 Claim 5 (canceled)

1 Claim 6 (currently amended): The method according to Claim ~~[[5]]~~ 1, wherein the using
2 ~~programmatically identifying step~~ further comprises ~~the step of consulting a lookup table, using~~
3 information regarding the content source when consulting the lookup component, ~~to thereby~~
4 ~~determine at least one element usable as a candidate value for the user to select when building the~~
5 ~~query command to query the content source:~~

1 Claim 7 (currently amended): The method according to Claim ~~[[5]]~~ 3, wherein the
2 ~~programmatically identifying step~~ further comprises ~~the step of consulting a lookup table, using~~
3 information regarding information regarding the user, ~~to thereby determine at least one element~~
4 ~~usable as a candidate value for the user to select when building the query command to query the~~
5 ~~content source~~ comprises at least one of: a role of the user, preferences of the user, a device used

6 by the user, or an identification of the user.

1 Claim 8 (currently amended): A computer-implemented method of programmatically building
2 queries, comprising steps of:

3 programmatically identifying, for each of at least one query parameter name to be used
4 when querying a content source, at least one candidate query qualifier by consulting a lookup
5 component using contextual information pertaining to a user, wherein each candidate query
6 qualifier specifies a comparator to use in determining a match for a value of that query parameter
7 name; and

8 displaying each of the programmatically identified query parameter(s) parameter names,
9 and for each query parameter name, each of the at least one candidate query qualifier(s) qualifiers,
10 on a user interface display configured ~~to allow a user to~~ build a query command, responsive to
11 input from the user, to query the content source, wherein the input from the user comprises
12 selecting ~~can select~~ at least one of the displayed query parameter(s) parameter names and, for
13 each selected query parameter name, one of the displayed candidate query qualifier(s), ~~to build the~~
14 ~~query command~~ qualifiers.

1 Claim 9 (currently amended): The method according to Claim 8, wherein the programmatically
2 identifying step further comprises ~~the step of consulting a lookup table~~, using information
3 regarding the content source when consulting the lookup component, ~~to thereby determine at~~
4 ~~least one element usable as a candidate query qualifier for the user to select when building the~~
5 ~~query command to query the content source.~~

1 Claim 10 (currently amended): The method according to Claim 8, wherein the ~~programmatically~~
2 ~~identifying step further comprises the step of consulting a lookup table, using contextual~~
3 ~~information regarding pertaining to the user, to thereby determine at least one element usable as a~~
4 ~~candidate query qualifier for the user to select when building the query command to query the~~
5 ~~content source; comprises at least one of: a role of the user, preferences of the user, a device~~
6 ~~used by the user, or an identification of the user.~~

1 Claim 11 (currently amended): A computer-implemented method of programmatically building
2 queries, comprising steps of:

3 obtaining a set of one or more query ~~parameters~~ parameter names for querying a content
4 source;

5 programmatically identifying, for the obtained set of query ~~parameters~~ parameter names,
6 one or more candidate extensions thereto ~~which are usable~~ for querying the content source by
7 consulting a lookup component using contextual information pertaining to a user, each of the
8 candidate extensions comprising an additional query parameter name for querying the content
9 source; and

10 displaying the set of query ~~parameters~~ parameter names, and the programmatically-
11 identified candidate extensions thereto, as an extended set of query ~~parameters~~ parameter names
12 on a user interface display configured ~~to allow a user~~ to build a query command to query the
13 content source[,]] responsive to selection, by the user, of wherein the user can select at least one
14 of the query ~~parameters~~ parameter names from the extended set to build the query command.

1 Claim 12 (currently amended): The method according to Claim 11, wherein the obtaining step
2 further comprises obtaining the set as input from ~~[[a]]~~ the user.

1 Claim 13 (original): The method according to Claim 11, wherein the obtaining step further
2 comprises programmatically determining the set.

1 Claim 14 (currently amended): The method according to Claim 11, further comprising the steps
2 of:

3 programmatically identifying at least one query extension parameter name for the query,
4 responsive to a request from the user to add at least one query parameter name to the set; and
5 displaying each of the programmatically-identified query extension ~~parameter(s)~~ parameter
6 names, in addition to the set of query ~~parameters~~ parameter names and the programmatically-
7 identified candidate extensions thereto, as the extended set of query ~~parameters~~ parameter names,

1 Claim 15 (currently amended): The method according to Claim 11, wherein the programmatically
2 identifying step further comprises ~~the step of consulting a lookup table~~, using information
3 regarding the content source; ~~to thereby determine at least one element usable as a candidate~~
4 ~~extension for the user to select when building the query command to query the content source~~
5 when consulting the lookup component.

1 Claim 16 (currently amended): The method according to Claim 11, wherein the programmatically

2 identifying step further comprises ~~the step of consulting a lookup table~~, using one or more of the
3 obtained query parameters parameter names when consulting the lookup component, ~~to thereby~~
4 determine at least one element usable as a candidate extension for the user to select when building
5 the query command to query the content source.

1 Claim 17 (currently amended): The method according to Claim 11, wherein the ~~programmatically~~
2 identifying step further comprises ~~the step of consulting a lookup table~~, using contextual
3 information ~~regarding~~ pertaining to the user comprises at least one of: a role of the user,
4 preferences of the user, a device used by the user, or an identification of the user, ~~to thereby~~
5 determine at least one element usable as a candidate extension for the user to select when building
6 the query command to query the content source.

1 Claim 18 (currently amended): The method according to Claim 11, further comprising the
2 [[step]] steps of:
3 selecting, by the user, at least one of the displayed query parameter names from the
4 extended set;
5 building the query command, responsive to the selecting; and
6 using the built query command, ~~built by the user by selecting at least one of the query~~
7 parameters from the extended set, to query the content source.

Claim 19 (canceled)

1 Claim 20 (currently amended): A system ~~[[for]]~~ configured to programmatically-building build
2 queries, comprising:

3 means for obtaining a set of one or more query-parameters parameter names for querying
4 a content source;

5 means for programmatically identifying, for the obtained set of query-parameters
6 parameter names, one or more candidate extensions thereto ~~which are usable~~ for querying the
7 content source by consulting a lookup component using contextual information pertaining to a
8 user, each of the candidate extensions comprising an additional query parameter name for
9 querying the content source; and

10 means for displaying the set of query-parameters parameter names, and the
11 programmatically-identified candidate extensions thereto, as an extended set of query-parameters
12 parameter names on a user interface display configured to ~~allow a user to~~ build a query command
13 to query the content source~~[[,]]~~ responsive to selection, by the user, of wherein the user can select
14 at least one of the query-parameters parameter names from the extended set ~~to build the query~~
15 ~~command~~.

1 Claim 21 (currently amended): A computer program product ~~[[for]]~~ configured to
2 programmatically-building build queries, the computer program product embodied on one or more
3 computer-readable storage media and comprising:

4 computer-readable program code for obtaining a set of one or more query-parameters
5 parameter names for querying a content source; and

6 computer-readable program code for programmatically identifying, for the obtained set of

7 query-parameters parameter names, one or more candidate extensions thereto ~~which are usable~~ for
8 querying the content source by consulting a lookup component using contextual information
9 pertaining to a user, each of the candidate extensions comprising an additional query parameter
10 name for querying the content source; and
11 computer-readable program code for displaying the set of query-parameters parameter
12 names, and the programmatically-identified candidate extensions thereto, as an extended set of
13 query-parameters parameter names on a user interface display configured to ~~allow a user to build~~
14 a query command to query the content source [[.]] responsive to selection, by the user, of wherein
15 ~~the user can select~~ at least one of the query-parameters parameter names from the extended set to
16 ~~build the query command.~~

1 Claim 22 (new): The system according to Claim 20, further comprising:

2 means for selecting, by the user, at least one of the query parameter names from the
3 extended set;

4 means for building the query command, responsive to the selecting; and

5 means for using the built query command to query the content source.

1 Claim 23 (new): The computer program product according to Claim 21, further comprising:

2 computer-readable program code for selecting, by the user, at least one of the query
3 parameter names from the extended set;

4 computer-readable program code for building the query command, responsive to the
5 selecting; and

6 computer-readable program code for using the built query command to query the content
7 source.